## REMARKS

This amendment is responsive to the Final Office Action dated September 7, 2007. Applicant has amended claims 1, 18, 35, and 47. Claims 1-54 are pending.

### Claim Rejection Under 35 U.S.C. § 103

In the Final Office Action, the Examiner rejected claims 1-5, 7, 18-22, 24, 47-51, and 53 under 35 U.S.C. 103(a) as being unpatentable over Drews (US 6,463,535) in view of Kozen "Efficient Code Certification". The Examiner also rejected claims 6, 23, and 52 under 35 U.S.C. 103(a) as being unpatentable over Drews in view of Kozen, and further in view of Rudoff et al. (US 6,263,378). Applicant notes the rejection of claim 2 based on Drews in view of Kozen and further in view Rudoff on page 4 of the Office Action most likely was a typographical error and instead assumes that the Examiner intended to direct this rejection to claim 6. Applicant respectfully traverses the rejection to the extent such rejections may be considered applicable to the claims as amended. The applied references fail to disclose or suggest the inventions defined by Applicant's claims, and provide no teaching that would have suggested the desirability of modification to arrive at the claimed invention.

With reference to independent claims 1, 18, and 47, for example, the applied references lack any teaching that would have suggested a device comprising an interface to retrieve boot code from a peripheral device upon power-up of the device and a memory module to store the boot code from the peripheral device, as required by Applicant's currently amended claim 18. The applied reference further lack any teaching that would have suggested the device further comprising a control unit to verify security of the boot code associated with the peripheral device by performing a security check on the boot code in accordance with a certificate that describes operation of the boot code, the control unit configured to execute the boot code to (i) initialize the peripheral device based on a result of the security check and (ii) subsequent to the initialization, provide an interface by which the control unit controls operation of the peripheral device, as further required by Applicant's currently amended claim 18.

Instead, the Drews reference begins by discussing problems related to information technology (IT) departments and the frequent time these departments spend traveling between jobs and remote facilities (Column 1, lines 21-38). Drews discloses that systems exist that attempt to eliminate a component of computers most likely to fail, e.g., a memory, and therefore most likely to cause IT departments to travel to varying facilities (Column 1, lines 45-60). The system described by Drews works in conjunction with these existing systems to ensure security within the existing systems, which previously were unsecure (Column 1, lines 53-67).

The existing system, Drews explains, reduces the amount of memory required in remote computers by enabling a server to provides a boot image during boot-up of the remote computers (Column 1, lines 45-52). The Drews system requires that the existing system provide the boot image as well as a manifest during a pre-boot operational state (Column 1, lines 61-67). Drews provides that a boot image refers to "an image of an application executed during the boot-up sequence of the local platform" (Column 2, lines 26-29). Drews further discloses that the boot image "may comprise one or more sub-images forming the entire boot image." (Column 3, lines 16-18) Drew also discloses that the manifest includes "(i) a secure hash value 300 for each sub-image of the boot image, (ii) a manifest digital signature 310, and (iii) a certificate chain 320 ..." (Column 4, lines 32-39) The Drews system therefore verifies that each sub-image of a boot image "has not been altered" based on hash values and digital signatures (Column 2, lines 32-35). To summarize, the Drews system provides for a verification module that utilizes the hash codes and digital signatures of the manifest to verify whether the downloaded boot image has been altered during a pre-boot operational state.

Applicant submits that the Examiner has misconstrued the Drews reference. Moreover, Applicant contends that the Examiner not only improperly construed Dews but further failed to properly construe Drews in view of Kozen. These arguments and other arguments are set forth below with respect to independent claim 18, however, as the Examiner acknowledged on page 2 of the Office Action by rejecting claims 1, 18, and 47 under the same rational, the arguments that follow apply equally to independent claims 1 and 47.

In rejecting Applicant's previously presented independent claim 18, the Examiner appears to suggest, contrary to the teachings of Drews, that central platform 110 constitutes a peripheral device (Office Action, pages 2-3). In elaborating on the rejection of an interface to retrieve boot code from a peripheral device upon power-up of the device, as required by Applicant's claim 18, the Examiner provides that column 3, lines 7-25 of Drews describes downloading a boot image and a signed manifest over a communications link (Office Action, pages 2-3). Drews, however, in column 3, lines 7-25 clearly describes that central platform 110 includes a server having at least one disk drive 115, and that local platform 120 may request a boot image to be downloaded over the communication link from the central platform. **A peripheral device of a computer is distinctly different from a central platform that provides boot images.** Examples of a peripheral device of a computer, as provided in paragraph [0032] of Applicant's pending application include a graphic device, a network controller and a storage controller, all of which contain different device drivers. A server, on the other hand, is an independent computing device that receives requests, as Kozen describes, and serves content, e.g., a boot image, to other computers.

The Examiner further misconstrues Drews by suggesting that Drews teaches that a control unit is configured to execute the boot code to *initialize the peripheral device* based on a result of the security check, as further required by Applicant's previously presented claim 18. As an initial matter, Applicant notes that **the Examiner, in rejecting both of claims 1 and 18, fails to address the initialization limitation** present in both of these independent claims. To establish a *prima facie* case of obviousness, MPEP 706.02(j) requires that the "prior art reference (or references when combined) must teach or suggest all the claim limitations." The Examiner, by failing to address this limitation, cannot establish a prima facie case of obviousness, and therefore Applicant, on this basis alone, requests prompt withdrawal of the rejections of claims 1, 18 and all claims dependent on these independent claims.

Drews fails to teach or suggest that a control unit is configured to execute the boot code to initialize the peripheral device, as required by Applicant's previously presented claim 18. Drews instead teaches a control unit that on the local computer is configured to execute the boot image to initialize <u>local platforms 120</u>, not the "peripheral device" (i.e., the server that provided the boot image according to the Examiner's logic).

Even assuming that the Examiner construes central platform 110 as an example of Applicant's peripheral device, Drews clearly makes no mention or reference to suggest executing the Drews boot image with local platforms 120 to initialize central platform 110, the server of central platform 110, or memory 115. Quite the contrary, Drews suggests that the boot procedure of the computer, e.g., local platforms 120, (i.e., the process by which a computer, such as local platforms 120, initialize components of that same computer) may be modified so that boot software is downloaded over a network, e.g., from central platform 110, and executed by the boot sequence (Column 1, lines 45-55). Therefore, as claims 1 and 18 **previously** presented this limitation and Drews moreover fails to disclose this limitation, Applicant request prompt withdrawal with respect to these claims. Applicant has amended independent claims 35 and 47 to include the initialization limitation, and therefore, also requests prompt withdrawal of the rejection with respect to these claims on the same basis.

Applicant, for purposes of clarity and unrelated to patentability, has amended independent claims 1, 18, 35, and 47. For example, independent claim 18, as currently amended, further requires that the control unit be configured to execute the boot code to provide, subsequent to the initialization, an interface by which the computer controls operation of the peripheral device. Drews, as described above with respect to the initialization limitation, does not disclose or even contemplate a control unit, such as local platform 110, configured to execute boot image 140 to provide an interface by which the computer controls operation of central platform 110, the server of central platform 110, or memory 115.

However, the Examiner, in rejecting claims 8, 25, and 54 under 35 U.S.C. 103(a) as being unpatentable over Drews and Kozen and in further view of England (US 6,757,824), addresses one particular type of boot code, i.e., a device driver, to initialize the peripheral device and define an application program interface for accessing and controlling the peripheral device, as required for example by Applicant's original claim 8. The Examiner suggested that column 7, lines 35-47 of England disclosed this control limitation and further provides that "England's method of initializing a device driver ... offers the advantage of ensuring that all components of the operating system are trusted and helps ensure that all storage will be secure (England, column 1 lines 50-65)." (Office Action, page 7)

Column 7, lines 35-47 however make no mention of initializing a device driver nor does Applicant comprehend what the Examiner means by "initializing a device driver." A device driver is not initialized rather a device driver is executed to initialize a peripheral. The Examiner's reason to combine England further mischaracterizes England. England does not require the components of the operating system to be trusted by the operating system, e.g., that these components are verified before being executed during the boot process of the local platform, but instead requires the components to be trusted by a separate content provider before the content provider will provide content to the local platform (Column 1, lines 50-65). **England therefore contradicts Drews, as they work directly opposite of one another.** Drews teaches to serving a boot image with a server to a local platform, while England teaches to uploading a boot certificate form a local platform to a server. Thus, these references should not be combined because the combined system provides no reasonable expectation of success, as MPEP 706.02(j) mandates the Examiner show to form a *prima facie* case of obviousness.

But even if combined, England fails to address the other deficiencies described herein with respect to Drews and Kozen, such as retrieving boot code from a peripheral device upon power-up of the device.

The Examiner has also misconstrued Drews in view of Kozen. Kozen discloses a method for performing Efficient Code Certification (ECC). The Kozen ECC method ensures control flow safety, memory safety, and stack safety (*See* bullet points on page 3) by generating a certificate during compilation of code, the combination of which is referred to as "certified code" (page 1, first paragraph). The certificate specifies structured annotations that direct a verification process executed by a verifier at load time (page 1, first paragraph and page 3, last paragraph). The Kozen reference explains that the verifier, guided by the structural annotations, "checks a simple set of conditions that are sufficient to imply the desired safety properties" (page 3, last paragraph). Kozen also provides that the Kozen ECC system is similar to JAVA in some respects (e.g., page 3, last paragraph) and suggests that a basic goal of this type of compilation and verification of code is to provide a simple and effective approach to thwart "critical security issues engendered by the rise of the Internet" (page 1, second paragraph). Therefore, the Kozen ECC system can fairly be said to ensure the above three safety aspects for application layer executables, similar to JAVA executables, by generating certified code that can be verified by a verifier to thwart critical security issues engendered by the rise of the Internet.

In rejecting Applicant's previously presented claim 18, the Examiner suggests that a person with ordinary skill in the art would utilize Kozen's method of verifying operation of a program because it offers advantages of ensuring "that executable code downloaded from an untrusted source is safe to run," but Applicant submits that the Examiner makes improper intuitive leaps in combining these two references. MPEP 706.02(j) states, in discussing 35 U.S.C. 103, that the Examiner should set forth in the Office Action "(C) the proposed modification of the applied reference(s) necessary to arrive at the claimed subject matter." The Examiner provides no such proposed modification by which one could arrive at the claimed subject matter, except to state that "Drew fails to teach a description of the operation of the boot code being verified" but that "Kozen teaches a description of the operation of the boot code being verified." (Office Action, page 3) Applicant contends for the following reasons that the Examiner has improperly combined these two references without providing a suitable "proposed modification."

First, the Kozen ECC method is directed to applications not boot code. JAVA is widely known to be run on-top of an operating system not during a pre-boot cycle. Kozen makes frequent references to JAVA, which suggests that the Kozen ECC method applies to applications, not boot code executed during a pre-boot process. Applications generally execute on top of an operating system, while pre-boot code executes prior to booting the operating system. The Examiner in combining Drews and Kozen seemingly assumes that the Kozen verifier can be executed during a pre-boot process, without providing any reference to Kozen that teaches pre-boot verification or even mentioning that such a modification to Kozen would be necessary. The Kozen verifier however most likely executes on top of an operating system, as there is no need to verify applications during a pre-boot process, as these applications cannot yet be loaded or cause security breaches. Drews suggests that this pre-boot verification is feasible, but only using techniques disclosed in Applicant's background, namely by way of digital signatures. Thus, the Examiner improperly assumes that combining the Kozen ECC system with the Drews verification system would reach the Applicant's claimed subject matter without further modification.

Second, the Kozen certificate is distinctly different from the Drews manifest. The Kozen certificate supplies the description of the operation of the boot code being verified. Because the Kozen certificate is generated during compilation of the boot code, the certificate necessarily pertains to a single application, not multiple applications. Yet, Drews explicitly teaches that a boot image may comprise multiple sub-images, and that for each sub-image the manifest includes a separate digital signature. The Examiner's wholesale combination of the Drews manifest and the Kozen certificate seems to suggest that the Kozen certificate replaces the Drews manifest. Under this assumption, Applicant submits that the boot image cannot include multiple sub-images, as each sub-image requires its own digital signature or, more generally, separate verification data. Thus, the combination of the Kozen certificate with the Drews manifest runs counter to the teaching of Drews, and the Examiner has improperly combined these two references by not providing a suitable "proposed modification."

The above arguments apply equally to independent claim 35, which was rejected under 35 U.S.C. 103(a) as being unpatentable over Drews in view of Kozen and Ong (US PGPub 2004/0177248). Claim 35 has been amended in substantially the same manner as claim 18, and therefore Applicant requests withdrawal of this rejection on the same basis stated above. Applicant further notes that the Examiner rejected dependent claim 40 under 35 U.S.C. 103(a) as being unpatentable over Drews in view of Kozen and Ong and further in view of England. As claim 40 depends on claim 35, this claim receives the benefit of the arguments made with respect to claim 35 and Applicant therefore similarly requests withdrawal of this rejection. All other dependent claims 2-8, 19-25, 36-41, 48-54 depend respectively on independent claims 1, 18, 35, and 47 and receive the benefit of the above arguments. Applicant therefore requests withdrawal of the rejection of these dependent claims.

For example, in rejecting claim 5, the Examiner suggests that Drews discloses in column 3, lines 7-25 that the Drews boot image includes boot firmware. Firmware, however, is distinctly different than an image of an application, which is the definition Drews provides in describing the disclosed boot image. Firmware, as its name suggests, is typically firmly embedded in a read-only memory (ROM), while an image of an application exists in a writeable portion of a memory module. Firmware is also generally recognized as being code necessary to initialize a device separate from the local platform or computer, such as a peripheral, while a boot image generally initializes the computer not a separate device. Drews therefore lacks any teaching or suggestion that a boot image includes boot firmware, and Applicant request withdrawal of the rejection to claims 5 and similar claims 22 and 51 for this reason and those reasons articulated above with respect to independent claim 1, 18, and 47.

For at least these reasons, the Examiner has failed to establish a prima facie case for non-patentability of Applicant's claims 1-54 under 35 U.S.C. 103(a). Withdrawal of this rejection is requested.

## CONCLUSION

All claims in this application are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of all pending claims. Please charge any additional fees or credit any overpayment to deposit account number 50-1778. The Examiner is invited to telephone the below-signed attorney to discuss this application.

Date:

December 7, 2007

SHUMAKER & SIEFFERT, P.A.
1625 Radio Drive, Suite 300
Woodbury, Minnesota 55125
Telephone: 651.735.1100
Facsimile: 651.735.1102

By:

Name: Kent J. Sieffert
Reg. No.: 41,312